

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1-34. (canceled)

35. (currently amended) A binary translation system, comprising:

a non-optimizing foreign code execution module configured to maintain dedicated foreign state for each foreign binary operation executed allowing for the exceptions arisen to be handled precisely; and

an optimizing binary translator configured to translate foreign binary operations into optimized sequences of host operations in such a way as to improve the speed of execution of the sequences; and

a host CPU configured to execute the host operations; and

a documentation generator configured to generate a set of documentations/documentation for optimized sequences of host operations, wherein ~~the each~~ documentation describes operations required to calculate a corresponding foreign state for the an appointed point;

~~a documentation tracker configured to track an executable path of optimized sequences of host operations and record host operation addresses at appointed points of the host operation sequences being executed, wherein, for each host operation address, operations~~ required to calculate a corresponding foreign state for the host operation address are added to documentation; and

a recovery mechanism configured to ~~use theselect~~ a documentation in the set of documentations using a host operation address corresponding to the selected documentation, to continue foreign codes execution based on information wherein the recovery mechanism is configured to perform the operations saved in the documentation to calculate the corresponding

22 | foreign state for the host operation address and to continue foreign codes execution in case of the  
23 | exception arisen during the execution of the corresponding optimized host codes.

1           36. (previously presented) The binary translation system of claim 35, wherein  
2 | the sequence of host operations consumes input values from the dedicated foreign state and puts  
3 | output results into the said foreign state.

1           37. (previously presented) The binary translation system of claim 36, wherein  
2 | the dedicated foreign state is not changed in case of exception arisen during execution of the host  
3 | operations sequence.

1           38. (previously presented) The binary translation system of claim 37, wherein a  
2 | foreign exception handler is executed in response to the exception occurring during execution of  
3 | the host operations sequence, the foreign exception handler configured to take input values from  
4 | the unchanged dedicated foreign state.

1           39. (previously presented) The binary translation system of claim 35, wherein  
2 | the optimizing translator is configured to combine all host operations sequences corresponding to  
3 | a number of foreign binary operations into one executable unit.

1           40. (previously presented) The binary translation system of claim 39, wherein  
2 | the optimizing translator removes intermediate outputs from the dedicated foreign state for each  
3 | sequence of host operations and saves information about the intermediate outputs in the  
4 | documentation.

1           41. (previously presented) The binary translation system of claim 40, wherein  
2 | the documentation describes a virtual foreign state between each sequence of host operations.

1           42. (previously presented) The binary translation system of claim 41, wherein  
2 | the optimizing translator preserves values from each virtual state in temporary registers without  
3 | re-using them until next virtual foreign state is passed.

1           43. (previously presented) The binary translation system of claim 42, wherein  
2 every memory write operation changes unconditionally the dedicated foreign state and is  
3 considered as a possible recovery point in case of exception.

1           44. (previously presented) The binary translation system of claim 43, wherein  
2 the recovery point corresponds to one of the virtual foreign states as described in the  
3 documentation.

1           45. (previously presented) The binary translation system of claim 44, wherein the  
2 documentation is a host to foreign registers computation rules table or host binary code used to  
3 recompute the foreign state.

1           46. (previously presented) The binary translation system of claim 35, wherein  
2 the host CPU is configured to execute host operations and produces exceptions as if it were a  
3 foreign CPU.

1           47. (previously presented) The binary translation system of claim 35, wherein  
2 documentation tracker is a special host operation, a recovery point register and an apparatus for  
3 saving instruction address in the recovery point register in response to executing the recovery  
4 point operation.

1           48. (previously presented) The binary translation system of claim 35, wherein  
2 the recovery mechanism is a host program configured to recompute a correct dedicated foreign  
3 state according to the documentation in the case of an exception generated.

1           49. (currently amended) A method of recomputing a dedicated foreign state in a  
2 binary translation system from documentation generated by an optimizing translator in a case of  
3 an exception arising during execution of optimized binary translated code translated from a  
4 foreign code, the method comprising:

5           designating a set of recovery points in the optimized binary translated code during  
6 optimized translation of the foreign code, wherein each recovery point represents a foreign state;

7                   generating a set of documentations during the optimized translation of the foreign  
8 code, wherein each documentation in the set of documentations corresponds to a recovery point  
9 in the optimized binary translated code and describes operations required to calculate a  
10 corresponding foreign state for the recovery point; and  
11                   using one of the documentations in the set of documentations corresponding to  
12 executed optimized binary translated code when an exception arises during its execution to  
13 recover a foreign state corresponding to a recovery point for the exception, wherein the foreign  
14 state is recovered by executing the operations for the one of the documentations.

1                   50. (previously presented) The method of claim 49, wherein the documentation  
2 describes a virtual foreign state and is discovered by using a recovery point register.

1                   51. (previously presented) The method of claim 50, wherein the optimizing  
2 translator sets recovery point operations in the host code in such a way that there is no  
3 unconditional changes in the dedicated foreign state between adjacent recovery points.

1                   52. (previously presented) The method of claim 51, further comprising:  
2 detecting an exception in the optimized binary translated code;  
3 reading the recovery point register;  
4 finding the appropriate documentation in the set of documentations corresponding  
5 to contents of the recovery point register;  
6 recomputing a foreign state according to the documentation;  
7 switching execution of the foreign code to non-optimizing execution;  
8 executing each foreign code operation in sequence;  
9 updating a dedicated foreign state at the end of each foreign code operation;  
10 detecting an exception during the course of non-optimizing execution; and  
11 transferring contents of the dedicated foreign state to the foreign exception  
12 handler.

1                   53. -56 (canceled)